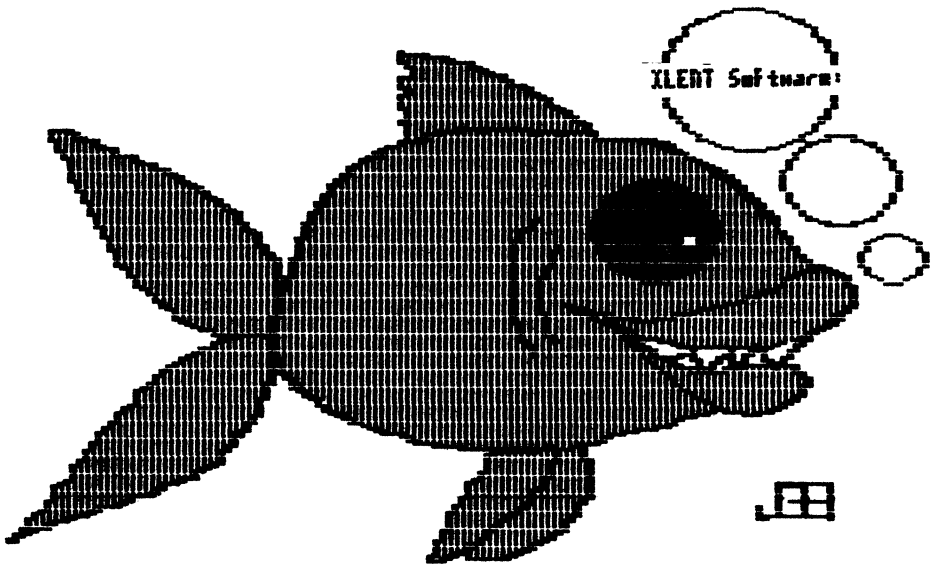
ATARI COMPUTER ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

MAY, 1985
Mike Dunn, Jim Bumpas & Larry Gold, Editors



MAY IS CATCH UP MONTH

News and Reviews

by Mike Dunn, Co-Editor

At our meeting in April, we had the wonderful opportunity to see and play with 2 new Ataris — the 128K 8-bit **130XE** shown by Computer Palace, and the 512K 16-bit **520ST**, thanks to Microbits. They do exist, are very nice, and the 130XE is now being sold for about $160 locally. The 520ST was a pre-production prototype that works, and is being sold to developers. A full report will be done by Kirt Stockwell, but I am impressed. The keyboards of both are the same, except the 130XE has fewer keys and no keypad. Both have a nice feel, and both units look solid and well built. The 520ST has very nice color resolution, about the same as an IBM-PC color monitor. The whole effect is like a color MacIntosh which is much quicker and slightly less resolution in color than the monochrome Mac (in monochrome, the Atari has better resolution than the Mac). Several members of the local Apple group were present, and their main reaction was disbelief. They said if Atari could produce them at the price they said, they would buy one! And these were the very avid Apple owners.

As a computer hobbyist for 6 years, I am not easily impressed. I own and use 2 Atari's, an IBM-XT, and a NEC lap computer; my wife has an Osbourne Exec and an ATR-8000 based CP/M system using mostly WordStar, and my two college daughters each have a system — one a ATR-8000 CP/M system using Perfect Writer and one an Atari system using Letter Perfect. I use my Atari the most, but the 520ST could easily become my favorite if I got one. We will keep you posted.

****

In the last issue, we had many programs and articles for the programs, but not enough room to list them all. This issue will have the rest of those listings. You will find **FID** by Dale Lutz, an ACTION! program for copying multiple files on a single disk drive as well as other utilites (very nicely done), **BlackBird** by Paul Freeman (beautifully done scrolling arcade game in BASIC similiar to Defender), UserComp from the UK Atari Group (a mini-assembler for creating User subroutes) and BIG by Carl Schwartz (a PILOT program to teach size value judgements to handicapped children). Remember, the articles to all the above programs were in the last issue (April '85).

The above programs (except the PILOT one, which will be on our next PILOT disk, ready soon) are on the new **Best of ACE #14**, as well as **ListPlus** by Paul Freeman allowing you to list inverse and graphic characters to an Epson printer. This is based on Greg Menke's program in an earlier ACE. The program by Dale LUtz (FID) is in ACTION! and run-time versions, and the DOS is the new **Atari DOS 2.5**, a must if you have an Atari 1050 drive, a 130XE. It includes a RamDisk and is configuable. Also new but not quite ready is the **KoalaPad Utility Disk**, a two sided disk with utility programs for the KoalaPad by Paul Freeman and a title screen generator with documtation by Bob Floyd from SPACE (St Paul (MN) ACE) — see the article by Graham Smith in the next issue.

****

The long awaited updated version of the **SoftFinder 2.1** is ready. It indexes from July '83 to Dec '84, and covers ANTIC!, ANALOG, COMPUTE! and our ACE Newsletter. As previous editions, it is available from Valley Soft, 2660 SW DeArmond, Corvallis, OR 97333, both on disk ($12)and printed form ($7). A very good reference.

For educators who want an easy, up to date reference to software for a wide number of computers, **the Digest of Software Reviews-Education** (301 W. Mesa, Suite F, Fresno, CA 93704, $124.95), this massive effort should meet your needs. Each program in described and exerpts of reviews from various publications are given (ACE is one of them!). You get about 700 pages of reviews in 5 monthly issues in a special storage binder.

****

ATR 8000 News

For those of you who have an ATR 8000 with a Co-Power 88, there is a new MS-DOS disk with an ANSI driver, communications patches, MS-DOS-CP/M transfer programs, etc. Send your original disk plus $25 to SWP's new address, 1000 W. Fuller, FT. Worth, TX 76115-3301. They also have 3 public domain disks for $10 each, #1 includes MEMBRAIN, a Ramdisk program for MS-DOS.

****

From Ralph Walden

The April ACE contained the same error in its description as the original error. The cross should be a "greater than" symbol (the "Insert" key). The typesetter prints a cross instead.

Since the release of the ACE-C compiler, there have been two updates correcting 3 bugs. The latest update available after the middle of April corrects a problem with multiplication. It seems if you multipy two numbers and the second number is negative, you'll get an incorrect answer. This version also corrects an earlier bug in the graphics() and locate() functions. If you have an early version of ACE-C and the graphics function doesn't work right, then pass the mode desired shifted left by 8 bits. The locate() function should be replaced with cgetc(6). (The new ST's use C in its GEM operating system, etc., so you can get a headstart by getting the ACE-C — Ed.)

Our sister club, A.C.E. N.S.W. (GPO Box 4514, Sydney 2001, NSW Australia) has produced cloth shoulder patches, approximately 3.5" in diameter. The patch is the "official" ACE symbol in cloth with blue, gold and red colors. Send $2 each (add $1 for airmail, or $2 airmail for orders in lots of 10) to ACE NSW, attention Alex Kwok, secretary-treasurer.

MAY PASSWORD
-F-A-I-R-
PHONE NO. 1-2-3-4

ATARI NEWS

The following is an official statement by Atari Corp. in response to concern over Atari's cancellation of booth reservations at this summer's Consumer Electronics Show in Chicago.

Since the new Atari Corp. was created, we have attempted to make a major statement in the personal computer industry. We have developed exciting new computer designs and a state-of-the-art machine in our ST computer technology. Because of this, and because we believe that none of the high-tech brand name computer makers will be at the Chicago consumer-oriented show, a marketing decision was made to re-direct our efforts toward more appropriate audiences.

ANTIC ON-LINE

(ANTIC PUBLISHING INC., COPYRIGHT 1985.
REPRINTED BY PERMISSION.)

250 CELEBRATE ANTIC ANNIVERSARY. SAM TRAMIEL FIELDS QUESTIONS — WORLDWIDE USERS NETWORK STARTS!

by NAT FRIEDLAND, ANTIC Editor

Users group officers from as far away as Pittsburgh, Boston and New Jersey attended the Worldwide Users Network inaugural meeting at Antic in San Francisco on March 30. There they heard Atari President Sam Tramiel say the company hopes to put users groups at the head of the line to buy ST 16-bit computers from the first production run .

Tramiel later explained to Antic that much detailed planning would be necessary before the complex process of setting users group ST distribution could be 100% certain. ANTIC ONLINE will report further information here as it becomes available.

ATARI OUT IN FORCE

Some 250 people attended Antic's combination WUN Meeting and gala Third Anniversary Party. The Atari delegation included Sam and Leonard Tramiel, Atarisoft President Sig Hartmann, Engineering Vice President Tom Brightman, Atari Explorer Editor Neil Harris and Users Group Coordinator David Duberman.

Atari ST Software Director Rich Frick ran a flashy demonstration program on Antic's own prototype 520ST. The demonstration featured a brand-new Forth cartridge which called up a sharply detailed series of digitized black and white photos — stored on a working Atari 10-megabyte 3.5" hard disk. XE Software Director John Skruch brought along a production 130XE and ran the animated robot demonstration (available on the SIG) which was shown at CES.

OTHER WELL KNOWN GUESTS

Joining the throng at Antic for oysters, champagne and birthday cake were Macintosh and IBM developers readying to produce ST software, as well as some of the top names in Atari third-party production. The guest list included Bill Wilkinson and Mark Rose of Optimized Systems Software, Alan Ackerman leading an MPP delegation that demonstrated the company's new hard disk, Rob LaTulipe of Digital Research (the makers of GEM) and Bob Moore of Hybrid Arts (MIDIMATE).

Mike Mock of Indus Disk Drives showed the upcoming (3rd quarter) $149.95 RAM-Charger card with 64K RAMdisk memory and a Z-80 processor that will work with a soon-to-come BIOS program to run CP/M software on the Indus GT and Atari.

Broderbund's representatives included the programmers of Print Shop, who got to read an advance copy of their Antic review from the upcoming June issue.

COMPUTER FAIRE

(A look at the new ATARI CORP.)
by Kirt E. Stockwell, past prez

I attended the West Coast Computer Faire as a representative of Microbits Peripheral Products, Inc., for whom I am the Director of Technical Support. The MPP booth was quite active (as we had hoped). The busiest booths, however, were those of two ATARI user groups. ATARI did not have a booth, but this did not stop them from enjoying the largest exposure and the most enthusiastic reactions of any manufacturer "represented" there. Two groups, the San Leandro Computer Club, (SLCC) and the Atari Bay Area Computer Users Society (ABACUS) managed to finagle booths side-by side. Since these were 10'X 10' locations, this gave them a storefront of 20 feet exposure. It wasn't enough!! Every time I stopped by the booths, the crowd was 5 deep ALL the way around the booths.

It is only faire (tee-hee) to mention that ATARI did have some official representation at th show: Officials from ATARI frequently stopped by the combined club booths, and the Corporation donated 5 to 10 thousand dollars worth of promotional materials to the clubs to be given away to a very interested public. ATARI also provided 2 of the new 520ST machines and several XE computers for display and demonstration. In addition, the SLCC recieved an ST system to be raffled off at the show. (A digression here: several other manufacturers also provided items to be raffled off by the SLCC. The income from the raffle was given entirely to an organization that provides assistance to dis-advantaged childen. Bravo!, SLCC). By cooperating with and assisting the user groups, ATARI CORP managed to turn about 10 thousand dollars worth of warehoused promo materials, plus an ST and a little respect, into about a half million dollars worth of publicity and enthusiasm.

The longer Jack Tramiel controls ATARI CORP., the more respect I gain for the man's intelligence and for his grasp of how to make limited dollars do incredible amounts of work. Any of you who have read my blatherings before should know I am an advocate of user groups in a big way. Before I became connected with MPP, I often said the USER GROUPS are a multi-million dollar resouce the manufacturers of hardware and software ignore to their own disadvantage. Since joining MPP, I have constantly lobbied for more and better user group support, and have been able to place some meaningful programs to benefit user groups. Of course, the benefits flow both ways!!! Tramiel, by taking the actions he did at the WCCF has gone a long way to legitimizing user groups in the eyes of the rest of the industry.

Other corporations with industry influence have lately begun making concerted, directed efforts at suppoting the User Groups. ANTIC magazine has sponsored the Worldwide User-group Network (WUN). ATARI is working closely with ANTIC in this endeavor. This will provide a single, focused point of dispersal for timely ATARI information as well as a single point to which ATARI can go for User Group opinions and feedback. This program was announced at the show, and was discussed in detail at a reception at ANTIC corporate offices. The list of BIG names at ATARI who attended the ANTIC/WUN reception includes practically all of the movers and shakers at ATARI. There is no doubt now as to whether ATARI is willing to support the user groups and treat them as worthwhile and responsible citizens of the computer world.

ATARI has introduced a computer which can truly be called a PERSONAL computer (based on price), yet will run rings around the two industry biggies, the IBM PC and the MACINTOSH/LISA. The speed alone is incredible. Couple this with COLOR, and the ability of using a high-res color monitor, SUPER high-res monochrome monitor, or your TV set, and you have a great system right off the bat. In addition, the disk drives are not only FAST, but have ½ meg and 1 meg (depending on which drive). The 512K RAM is not eaten up with FONTS, OPERATING SYSTEMS and junk. There is 192K of ROM in the ST. This is where the OS, GEM, and the FONTS reside, as well as whichever "native" language will come with the machine. This leaves your 512K virtually untouched and free for application software.

The keyboard is the nicest I've felt since I test drove a DEC Professional PC. In fact, if any computer has a nicer keyboard, I haven't seen it yet. The case is beautiful, and the set-up is — well, the set-up is pure ATARI. Detractors will make loud rude noises about the separate power supplies and "ALL THOSE CABLES" that are, indeed, a part of the system. Instead of a lengthy lecture on the design constraints of a personal computer system, let me say there really was no other alternative except to build another computer looking like the IBM PC and being considerably more expensive. ATARI considered this to be unacceptable and decided if you are after POWER (without the price) you could buy an ST. If you are after STATUS or IMAGE, you have the IBM PC and the MACINTOSH family.

SOFTWARE: ATARI will gladly sell a development system to anybody who is willing to plop down the cash up front at a minimum of FULL RETAIL. In addition, developers will have to take seminars provided by Digital Research, and will have to pay for the development system software. (I estimate that MPP has spent about $8,000.00 to acquire the necessary equipment, tools, and information.) Using this technique, ATARI can realistically make plans based on WHO IS REALLY DEVELOPING SOFTWARE FOR THE ST (and there is no shortage of software developers working on programs NOW).

Will I buy an ST? You bet! Not only will I buy one when they a- available, but I have already recommended them to several seri seekers of computer power, and will continue to do so. Am I impres. by the new machines? Not only am I impessed by the new machines, but also by the fact that ATARI is getting them out ON TIME.

While attending the WCCF, I had an opportunity to visit a general meeting of the SLCC as a guest speaker. I was and am impressed by the caliber of the membership and leadership of that group. For a local club, they are top notch.

MEGAFONT II+

Megafont II+ ($25, XLEnt Software, Box 5228, Springfield, VA 22150) is a new version of an old program, but it is more than that. It is an updated version of a very good program so people with printers who could not take advantage of this program before can now do so. New features are also added to allow the user to do more things than ever.

Owners of Epson and the various related printers can now have four different size screen dumps. The left margin can be adjusted to allow for logos or pictures to be used in letterheads or whatever you wish. At the same time Megafont II+ allows you to pause the printer so you won't overheat it when dumping a picture etc.

Another of the features added is the Fast Print option. This is added to speed up the time it takes to list out programs. I like this feature since printing out the program listings is one of the things I do on the newsletter and it takes a long time to print out the listings for some programs.

Another new feature is called the Font Splicer which allows you to combine two fonts together. With certain word processors, Atariwriter, Bank Street Writer, Text Wizard, etc. you can take out the control codes and print your files in any font which Megafont II+ supports using Megafont's controls to give you a new type of printing of your files.

There are many things one can do with this program, including dumping mode 7+ and 8 graphics. Koalapad and Atari Artist screens can be printed with Megafont II+.

I've started using this program and find it not only does what I want, but it gives me the flexibility to change things around the way I w- them and in turn enhance my printing to make a better looking pag whatever I am doing. I can only say the people at XLEnt Software t- a good program and made it better than ever. The instructions are clear and concise and not too long. It is a menu driven program so one should be able to use Megafont II+ within a matter of minutes.

— Larry Gold

VP RAMBLINGS

Here it is the month of May and all kinds of good things are happening. The 520ST was demostrated at our last meeting and it looks super and runs the same. From what we have heard as rumors and such delivery should be right about now. I sure hope they keep their (Atari) schedule.

We are changing the way we use the BBS, from now on we will print a new password each month and when you log on when the system asks for your password type in the one given in the newsletter. When it asks for the last four numbers of your phone number type in 1234. If you haven't received your newsletter yet you still can use last month's password until the middle of the month when it will be deleted. Non-members will have their own password which will allow access to the lower levels of the system as before. I hope this new system works. The change is needed because this BBS has problems when the number of active passwords exceeds 600 or so. Please let us know what you think of it and perhaps any other ways to improve the system.

Check the newsletter for the new password.

We need more imput from you to let us know what you want to see in the newsletter, especially with the new machines coming out. We hope to expand so we have programs and articles covering all the computers and languages. If you don't write you may not get what you want.

— Larry Gold, Vice President

MAY PASSWORD IS FAIR

PHONE NO. IS 1234

MOVING ALONG IN LOGO

(The SHAPE EDITOR and Simple Animation)
by Ruth Ellsworth

Giving the Turtle a new look with the SHAPE EDITOR can be a lot of fun. Trying to save and reload shapes can be not so fun. The reason for this is the LOGO documentation does not give clear enough or specific enough information on the procedures necessary.

The easiest way to save and reload shapes is to create a Save(shape) module and a Load(shape) module. For example:

```
TO SAVEDOG
MAKE "DOG1 GETSH 1
MAKE "DOG2 GETSH 2
(etc.)
END

TO LOADDOG
PUTSH 1 :DOG1
PUTSH 2 :DOG2
(etc.)
END
```

The program is saved to disk or cassette by what ever name is desired In order to retrieve the shapes, the program is loaded as usual. The LOADDOG module must be called before the program is run in order to load the shapes into the SHAPE EDITOR. This method of saving and loading shapes is the easiest and requires least typing. On our computer this method is not reliable for more than two different shapes.

To prevent problems with multiple shapes, the PUTSH command can be used. The TO LOADDOGS module in the program listing uses this method of saving and loading shapes. In order to use the PUTSH command one must place 16 numbers which represent the bits in the shape wanted in the brackets which follow the PUTSH command. This may sound complicated, but can easily be accomplished by using the EDSH command to create the shape, then using the PR GETSH commands to return the bit numbers in the shape. Note again that in order to have the shape in the SHAPE EDITOR, one must call the LOAD(shape) module after the program is loaded before the program is run. In the program listing LOADDOGS is called by both the BEGIN and the START modules to insure the presence of the shapes before either of the two demo programs is run.

The GODOG program is a demonstration of simple animation. In this case, four dog shapes are used to move the turtle in the shape of a dog across the screen. Like a "flip movie" made out of paper or cartoon, each dog is drawn in a slightly different position. The shapes are run in rapid order to give the appearance of moving. The WAIT command is used to slow the movement down enough to be seen. When the WAIT command is lengthened, it is easier to see the movement from one "frame" (picture) to another.

To run the GODOG demo, type START. When the dog appears, type G to make the dog move. The dog will move to the right through each of the shapes which give the illusion of running. Then it will pause. Type G to continue or S to return it to the initial. Use the BREAK key to stop this demo which is what my children call "a forever program."

The second demo demonstrates the use of the joystick to move shapes around the screen. There are eight dog shapes used, and younger children enjoy this one. It is particularly fun if dog like sounds are added. To use the demo, type BEGIN. The BEGIN module loads the dog shapes, sets the beginning dog shape, and waits for the WHEN demon to show that the joystick has been moved. If the value of the joystick is less than 0, the joystick is not being moved and action stops. The position of the joystick determines which series of dog shapes are used; they appear in the brackets after the joystick number. IF (JOY 0) = 0 [DOGUP] means that IF the JOYstick in the first port returns a 0, the DOGUP module is called.

The SETSP and RANDOM commands can be used with shapes to really "move graphics programs" along. To changes the GODOG module so that it demonstrates their use, use SETSP in place of the FD lines and RANDOM degrees in the place of the RT command.

MAY MEETING WED. MAY 8TH
SOUTH EUGENE HIGH 7:30 PM

F-15 STRIKE EAGLE

(Microprose Software $34.95)

F-15 Strike Eagle is a simulation putting you in command of a modern jet fighter-bomber. At your fingers are medium and short range missles, guns and bombs. Ahead of you are SAMs, enemy fighters, airfields, and targets. The mission is to penetrate the enemy defenses and bomb your primary target(s). Are you good enough?

This is a change for me. Usually I play ground combat simulations, but I never turn away a challenge. And this game is a challenge, for pilots of all ages. The game disk includes seven missions to fly, from Libya to Hanoi. One to four players may participate. There are also four levels of difficulty, and they are difficult. Realism plays a big part in this game. Before you can begin a mission, you must authenticate yourself with a code found in the book. More realism exists in the game as your plane is equipped with everything from HUD (Heads up Display, showing instrument read-outs on the windscreen) to flares to ECM (Electronic Counter Measures to defeat enemy weapons). You look out from the cockpit over the landscape as you fly, but your dash board includes three radars, as well as a tactical map.

This game is very playable, and flexible in that no mission is exactly the same as any other. The graphics are very detailed, along with power/weight curves and thrust output graphs included in the manual, and even diagrams of some simple evasive actions. The aircraft is so maneuverable you can perform almost any type of maneuver. I rate this game highly, for the true pilot or just for someone who wants to enjoy the thrill of flying.

—Aaron Ness

1030 RING DETECT

(reprint: Computer Squad News, March 1985)

If you've ever thought about running a BBS on a 1030 modem, here is some information you may need to know. There are only 4 parts required to build a ring detector: 1) A phone wire to connect to your phone line. The plug-in type is best. 2) A Joystick replacement cable. 3) A 125 volt AC relay. Radio Shack #275-217B or equivalent. 4) A 22 MF or greater capacitor of at least 100 volts. 5) An optional plastic box. DO NOT USE METAL.

Connect the Green Phone Wire to pin 7 of the relay coil.

Connect the + (positive) end of the 20MF capacitor to pin 8 of the relay coil.

Connect the Red Phone Wire to the – (negative) end of the capacitor.

Connect the unit to the phone line and test to see that it responds. REMOVE IT FROM THE PHONE LINE!!

Connect the Orange Joystick wire to the Common Relay Contact pin 4.

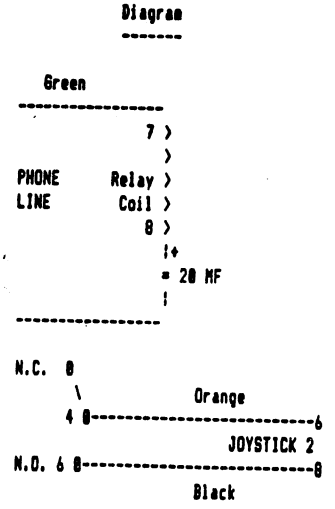
Connect the Black Joystick Wire to the N.O. Relay Contact pin 6.

With a meter or circuit tester, check that no connection between the phone wires and the joystick wires occurs. Cross check all four wires.

Package up, to protect the unit and install in Joystick port 2.

Plug the unit into the phone jack. You may need a Y telephone adapter jack.

You are now ready to run your software.



FID BY DALE LUTZ

```
;           FID
;       by Dale Lutz


;This program is a single drive
;utility for file manipulations.
;It excels in duping a number of small
;files, eg. music composer files.
;It could also be used in mass file
;density conversion with a single
;Indus Drive.
;The scrolling directory routines
;should be able to be incorporated into
;other ACTION! programs with a minimum
;of effort.
;The codetop constant is set to the top
;of the program and serves as a pointer
;to bottom of the buffer used for
;file copying.  It may need to be
;changed depending on your memory config
uration
;Right now it is set so that when
;it is comiled off the disk, and
;SYS.ACT is INCLUDED, and OS/A+
;is in memory, it is okay.  To find
;out what to change it to, use the
;clear memory, Compile this from disk,
;and then type ?$E from the monitor.
;Note the last number typed, and change
;the codetop to this value+200...Good lu
ck

; INCLUDE"SYS.ACT" ;used when making it
runtime

DEFINE length="18",
       codetop="19500" ;CHANGE THE 19500
 TO THE
               ;VALUE RETURNED AT THE EN
D OF COMPILING
               ;BY ?$E

BYTE ARRAY directory(1170),mark(65),file
name(20)
BYTE counter,consol=53279,choice,formatt
ed,flag
INT pos,y
CARD base,c,memtop=741
BYTE POINTER buffer
CARD ARRAY msg(7),addr(66)

PROC Menu()
  BYTE a
  msg(2)="LOCK FILES"
  msg(3)="UNLOCK FILES"
  msg(4)="DELETE FILES"
  msg(1)="COPY FILES"
  Graphics(0)
  Poke(752,1)
  Color=160 ;draw frame
  Plot(0,0)
  DrawTo(39,0)
  DrawTo(39,23)
  DrawTo(0,23)
  DrawTo(0,0)
  Plot(0,0)
  Position(10,23)
  Print("Written In ACTION!")
  Position(10,1)
  PrintE("Atari File Developer")
  Position(10,2)
  FOR a=1 to 20
    DO
    Put(13)
    OD
  Position(14,3)
  PrintE("Version 1.4")
  Position(18,5)
  PrintE("by")
  Position(14,7)
  PrintE("Dale Lutz")
  Position(8,9)
  PrintE("Functions Available:")
  Position(12,11)
  PrintE("1.  Batch Copy Files")
  Position(12,13)
  PrintE("2.  Lock Files")
  Position(12,15)
  PrintE("3.  Unlock Files")
  Position(12,17)
  PrintE("4.  Delete Files")
  Position(12,19)
  PrintE("5.  Format Disk")
  Position(12,21)
  Print("Your Choice-->")
  Put(160)
RETURN

PROC Blip()
  BYTE count=[0]
  IF count>4 THEN count=0 FI
  Position(12,11+count*2)
  Put('1+count)
  count==+1
  IF count>4 THEN count=0 FI
  Position(12,11+count*2)
  Put('1+128+count)
RETURN

PROC Format(BYTE ARRAY DriveSpec)
  XIO(5,0,254,0,0,DriveSpec)
RETURN

PROC Erase()
  XIO(5,0,33,0,0,filename)
RETURN

PROC Protect()
  XIO(5,0,35,0,0,filename)
RETURN

PROC UnProtect()
  XIO(5,0,36,0,0,filename)
RETURN

PROC Read_Directory()
  BYTE ARRAY str(30)
  counter=0
  base=directory
  Close(1)
  Open(1,"D1:*.*",6,0)
  WHILE EOF(1)=0
    DO
    InputSD(1,str)
    counter==+1
    SCopy(base,str)
    base==+length
    OD
  Close(1)
RETURN

PROC Markprint(BYTE POINTER begin,BYTE p
lace)
  BYTE POINTER b
  BYTE chr
  FOR b=begin+1 to begin^+begin
    DO
    c=pos+place
    chr=b^+128*mark(c)
    Put(chr)
    OD
  PutE()
RETURN

PROC Updatey()
  Position(length+1,y)
  PrintE("<-")
  IF y<23 THEN
    Position(length+1,y+1)
    PrintE("  ")
  FI
  IF y>0 THEN
    Position(length+1,y-1)
    PrintE("  ")
  FI
```

```
  consol=7
RETURN

PROC Dirprint()
  CARD a
  BYTE end
  Position(2,1)
  IF counter-2)19 THEN end=19
  ELSE end=counter-2
  FI
  FOR a=0 TO end
    DO
    base=directory+(pos+a)*length
    Markprint(base,a)
    OD
RETURN

PROC Downdir()
  Position(2,1)
  Put(157)
  base=directory+pos*length
  Markprint(base,0)
  Position(2,21)
  Put(156)
RETURN

PROC Updir()
  Position(2,1)
  Put(156)
  Position(2,20)
  base=directory+(pos+19)*length
  Markprint(base,19)
RETURN

PROC Up()
  y==-1
  IF y=0 THEN y=1
    IF pos)0 THEN pos==-1
      Downdir()
    FI
  FI
  Updatey()
RETURN

PROC Down()
  y==+1
  IF y)counter-2 THEN y==-1 FI
  IF counter=22 AND pos=0 THEN
    IF y=20 THEN pos==+1 y=20
      Updir()
    FI
  FI
  IF y=21 THEN y=20
    IF pos<counter-20 THEN pos==+1
      Updir()
    FI
  FI
  IF y=20 AND pos=counter-21 THEN y=19 F
I
  Updatey()
RETURN

PROC Command()
  BYTE key=764,oldpos,a
  oldpos=pos
  IF key=14 THEN Up()
  ELSEIF key=142 THEN
    FOR a=1 TO 5
    DO
    Up()
    OD
  ELSEIF key=15 THEN Down()
  ELSEIF key=143 THEN
    FOR a=1 TO 5
    DO
    Down()
    OD
  ELSEIF key=12 OR key=140 THEN
    y==-1
    a=pos+y
    IF mark(a)=0 THEN mark(a)=1
    ELSE mark(a)=0
    FI
    c=directory+a*length
    Markprint(c,y)
    y==+1
    IF y<20 AND y+pos<>counter-2 THEN y=
=+1 FI
    Updatey()
  FI
  key=255
RETURN

PROC More_update()
  BYTE xp=84,yp=85,ox,oy
  oy=yp
  ox=xp
  Position(6,0)
  IF pos<>0 THEN
    Put(27) Put(28)
    Print("MORE")
    Put(27) Put(28)
    Print("      ")
  ELSE Print(msg(choice))
  FI
  Position(6,21)
  IF counter-21<>pos AND counter-2)19 TH
EN
    Put(27) Put(29)
    Print("MORE")
    Put(27) Put(29)
    Print("      ")
  ELSE Print(msg(choice))
  FI
  yp=oy
  xp=ox
RETURN

PROC Get_selections()
  CARD j
  FOR j=0 TO 65
    DO
    mark(j)=0
    OD
  Graphics(0)
  Poke(752,1)
  Read_Directory()
  pos=0 y=1
  Dirprint()
  Updatey()
  WHILE consol<>6
    DO
    Command()
    FOR j=1 TO 100 DO OD
    More_update()
    OD
RETURN

PROC Get_filename(BYTE POINTER begin)

  ;enter with begin pointing to correct
  ;string exit with string filename read
y to go

  BYTE ARRAY te
  BYTE POINTER b
  BYTE chr,co,a
  co=0
  FOR b=begin+3 to begin+10
    DO
    chr=b^
    IF chr=32 THEN EXIT
    FI
    co==+1
    OD
  filename(0)=co+4
  a=0
  FOR b=begin+3 TO begin+co+2
    DO
    a==+1
    filename(a)=b^
    OD
  a==+1
  filename(a)='.
  FOR b=begin+11 TO begin+13
```

```
    DO
    a==+1
    filename(a)=b^
    OD
  SCopy(te,filename)
  filename(0)==+2
  filename(1)='D
  filename(2)=':
  FOR a=1 TO te(0)
    DO
    co=a+2
    filename(co)=te(a)
    OD
  PrintE(filename)
RETURN

PROC Execute()
  BYTE loop,sp
  Get_selections()
  Put(125)
  c=msg(choice)
  FOR loop=1 TO (38-PEEK(c))/2
    DO
    Put(32)
    OD
  PrintE(c)
  PutE()
  FOR loop=0 to counter
    DO
    IF mark(loop)=1 THEN
      c=directory+length*loop
      Get_filename(c)
      IF choice=2 THEN Protect()
      ELSEIF choice=3 THEN UnProtect()
      ELSEIF choice=4 THEN Erase()
      FI
    FI
    OD
RETURN

PROC MyError()
  formatted=0
RETURN

PROC Checkformat()
  BYTE ARRAY str(30)
  BYTE loop
  CARD temperror,striing
  Graphics(0)
  striing="Checking Disk"
  FOR loop=1 TO (38-Peek(striing))/2
    DO
    Put(32)
    OD
  PrintE(striing)
  PutE()
  formatted=1
  temperror=Error
  Error=MyError
  Close(1)


;Remove the following line if you don't
;want to wait while it checks if a disk
;is formatted:

  Open(1,"D1:*.*",6,0)

  Close(1)
  Error=temperror
  IF formatted=1 THEN
    PrintE("This disk is already formatted.")
    Print("Enter 'Y' to FORMAT it anyway-->")
    InputS(str)
    IF str(1)<>'Y AND str(1)<>'y THEN
      RETURN
    FI
  FI
  PutE()
  PrintE("Formatting Disk...")
  Format("D1:")
RETURN

BYTE FUNC GetKey()
  BYTE number
  Close(2)
  Open(2,"K:",4,0)
  number=GetD(2)
  Close(2)
  Put(number)
RETURN (number)

PROC Dump(BYTE a)
  BYTE loop,end,number
  PutE()
  Put(253)
  Print("Insert Destination Disk, Press RETURN")
  Print("              -->")
  Poke(764,255)
  number=GetKey()
  FOR loop=0 to a-1
    DO
    IF mark(loop)=1 THEN
      c=directory+length*loop
      Print("Writing File:  ")
      Get_filename(c)
      Close(3)
      Open(3,filename,8,0)
      IF SCompare("D:DOS.SYS",filename)=0 THEN
        PutDE(3)
      ELSE end=loop+1
        FOR buffer=addr(loop) TO addr(end)-2
          DO
          number=buffer^
          PutD(3,number)
          OD
      FI
      Close(3)
      mark(loop)=0
    FI
    OD
  IF a<65 THEN
    PutE()
    Put(253)
    Print("Insert Source Disk, Press RETURN")
    Print("              -->")
    Poke(764,255)
    number=GetKey()
  FI
RETURN

PROC Readfile(BYTE a)
  BYTE number
  Close(4)
  Open(4,filename,4,0)
  addr(a)=buffer
  WHILE EOF(4)=0
    DO
    number=GetD(4)
    buffer^=number
    buffer==+1
    IF buffer=memtop THEN
      Dump(a)
      buffer=codetop
      flag=1
      RETURN
    FI
    OD
  number=a+1
  addr(number)=buffer
RETURN

PROC Copy()
  BYTE a
  flag=0
  Get_Selections()
  buffer=codetop
  FOR a=0 TO 65
    DO
```

BLACKBIRD

```
2 REM * BLACKBIRD *
3 REM * Paul Freeman *
4 REM * 8-25-84 *
9 POKE 106,PEEK(106)-5
10 CLR :GOTO 7000
15 H=H+5:V=V+5:CR$=D1$:RETURN
16 H=H+5:V=V-5:CR$=D1$:RETURN
17 H=H+5:CR$=D1$:RETURN
19 H=H-5:V=V+5:CR$=D2$:RETURN
20 H=H-5:V=V-5:CR$=D2$:RETURN
21 H=H-5:CR$=D2$:RETURN
23 V=V+5:RETURN
24 V=V-5:RETURN
25 V=V+1:RETURN
100 POKE 53278,1:GOSUB PEEK(632)+10:P0$(V)=CR$:POKE 53248,H
103 IF PEEK(764)<>255 THEN GOTO 23000
105 POKE 53764,130-LV
110 IF PEEK(53252)<>0 OR PEEK(53260)<>0 THEN GOSUB 800
120 IF H>190 THEN H=60:SCR=SCR+1:FR=0:M$(MV)=B$:P2$(RV)=B2$:SC=SC+30:GOSUB 10500+SCR*500:GOTO 410
130 IF H<60 THEN H=190:SCR=SCR-1:FR=0:M$(MV)=B$:P2$(RV)=B2$:SC=SC-30:GOSUB 10500+SCR*500:GOTO 410
200 IF PEEK(644)=1 AND FR=0 THEN GOTO 400
210 IF FR=0 THEN FR=1:MH=H+3:MV=V+4:POKE 53252,MH
220 MV=MV+M5:M$(MV)=DM$:IF MV>215 THEN CL=0:GOTO 240
230 CL=PEEK(53248):IF CL=0 THEN GOTO 400
240 IF CL<>2 THEN FR=0:M$(MV)=B$:MH=H+3:MV=V+4:GOTO 400
245 FR=0:POSITION 14,23:SC=SC+40+INT(V/20):? #6;SC:POKE 712,62
250 M$(MV)=B$:POSITION INT(MH/8-4.865)-1,INT(MV/8-3.645)-0:? #6;" ":MH=H+3:MV=V+4:POKE 712,0
399 REM * ROCKET *
400 IF PEEK(53254)=0 AND RH<200 AND RH>50 THEN GOTO 450
410 P2$(RV)=B2$:RP=INT(6*RND(1)):RH=RI(RP+6*(SCR-1),0):RV=RI(RP+6*(SCR-1),1):POKE 53250,RH
420 POKE 706,(INT(16*RND(1))*16)+10
450 RH=RH+SGN(H-RH)*2:RV=RV-R5:P2$(RV)=D4$:POKE 53250,RH:GOTO 100
799 REM * COLLISION *
800 P2$(RV)=B2$:POKE 53761,0:POKE 53765,0:IF LV=0 THEN GOTO 22000
803 FOR A=0 TO 10:POKE 712,14:POKE 704,(2*16)+10-A:V=V+1:POKE 53256,1:P0$(V)=D3$:FOR W=1 TO 10:NEXT W:P0$(V)=D1$
805 POKE 712,0:POKE 53760,50+A*15:POKE 53761,14-A:FOR W=1 TO 5:NEXT W:POKE 53256,0:P0$(V)=D2$:FOR W=1 TO 5
806 NEXT W:NEXT A
808 SOUND 0,0,0,0:POKE 53768,4:POKE 53761,162:POKE 53765,162:POKE 53760,252:POKE 53764,130-LV
810 P0$(V)=B$:H=65:V=110:CR$=D1$:POKE 704,15:POKE 53256,0:LV=LV-1:POSITION LV,23:? #6;" "
830 RP=INT(5*RND(1)):RH=RI(RP+5*(SCR-1),0):RV=RI(RP+5*(SCR-1),1):POKE 53250,RH:POKE 53278,1:RETURN
6999 REM * GR SETUP *
7000 GRAPHICS 1+16:SETCOLOR 0,3,6:SETCOLOR 2,8,4:SETCOLOR 1,0,10:SETCOLOR 3,8,12
7499 REM * VARIABLES *
7500 H=70:V=110:S=5:M5=4:SCR=1:SC=0:FR=0:MH=H+3:MV=V+4:LV=5:SKL=1:R5=3
7999 REM * P/M *
8000 DIM F1$(1),F2$((INT(ADR(F1$)/2048)+1)*2048-ADR(F1$)-1)
8010 DIM B$(256),F3$(256),F4$(256),M$(256),P0$(256),P1$(256),P2$(256)
8020 B$=CHR$(0):B$(256)=CHR$(0):B$(2)=B$
8025 P0$=CHR$(0):P0$(256)=CHR$(0):P0$(2)=B$
8028 P2$=CHR$(0):P2$(256)=CHR$(0):P2$(2)=B$
8030 F3$=B$:F4$=B$:M$=B$:P1$=B$:P2$=B$

8040 DIM D1$(18),CR$(18):RESTORE 8060:FOR I=1 TO 16:READ A:D1$(I,I)=CHR$(A):NEXT I:CR$=D1$
8060 DATA 0,0,0,0,0,128,194,127,120,224,128,0,0,0,0,0
8070 DIM D2$(18):RESTORE 8080:FOR I=1 TO 16:READ A:D2$(I,I)=CHR$(A):NEXT I
8080 DATA 0,0,0,0,0,1,67,254,30,7,1,0,0,0,0,0
8085 DIM D3$(18):RESTORE 8090:FOR I=1 TO 14:READ A:D3$(I,I)=CHR$(A):NEXT I
8090 DATA 0,0,0,128,10,160,86,40,129,80,0,0,0,0
8100 RESTORE 8110:DIM DM$(13):FOR I=1 TO 9:READ A:DM$(I,I)=CHR$(A):NEXT I
8110 DATA 0,0,0,0,2,0,0,0,0
8120 DIM D4$(20):RESTORE 8130:FOR I=1 TO 15:READ A:D4$(I,I)=CHR$(A):NEXT I
8130 DATA 8,8,8,8,28,8,28,62,42,0,0,0,0,0,0
8140 DIM B2$(30):B2$=B$:POKE 54279,ADR(B$)/256:POKE 559,62:POKE 53277,3:POKE 53256,0
8200 POKE 53248,H:POKE 623,2:GOSUB 20000
9999 REM * CHAR SET *
10000 DIM ML$(25):ST=(PEEK(106)+1)*256

10025 RESTORE 10030:POKE 204,ST/256:POKE 206,224:FOR A=1 TO 20:READ B:ML$(A,A)=CHR$(B):NEXT A:A=USR(ADR(ML$))
10030 DATA 104,162,4,160,0,177,205,145,203,200,208,249,230,206,230,204,202,208,242,96
10040 POKE 756,ST/256:FOR A5=65 TO 83:POKE (A5-32)*8+5T+7,255:POKE (A5-32)*8+ST,255:NEXT A5
10042 RESTORE 10043:FOR B=0 TO 7:READ C:POKE (77-32)*8+5T+B,C:NEXT B
10043 DATA 255,0,0,0,0,0,0,255
10045 FOR B=1 TO 6:POKE (80-32)*8+5T+B,PEEK((82-32)*8+5T+B):NEXT B
10047 FOR B=1 TO 6:POKE (78-32)*8+5T+B,PEEK((83-32)*8+5T+B):NEXT B
10048 FOR A=0 TO 6 STEP 2:POKE (35-32)*8+5T+A,170:POKE (35-32)*8+5T+A+1,85:NEXT A
10050 RESTORE 10060:FOR A5=81 TO 90:FOR B=0 TO 7:READ D:POKE (A5-32)*8+5T+B,D:NEXT B:NEXT A5
10060 DATA 0,136,153,153,219,251,251,255
10065 DATA 255,251,251,219,153,153,136,0
10070 DATA 0,34,35,103,103,119,247,255

10075 DATA 255,247,119,103,103,35,34,0

10080 DATA 255,147,255,201,255,147,255,201
10085 DATA 255,255,255,255,255,255,255,255
10090 DATA 0,144,144,216,218,250,255,255
10095 DATA 255,127,111,47,15,13,5,0
10100 DATA 0,5,13,15,47,111,127,255
10105 DATA 255,255,250,218,216,144,144,0
10110 RESTORE 10120:FOR B=0 TO 7:READ D:POKE (74-32)*8+5T+B,D:NEXT B
10120 DATA 195,195,102,126,24,60,165,255
10130 RESTORE 10140:FOR B=0 TO 7:READ D:POKE (42-32)*8+5T+B,D:NEXT B
10140 DATA 0,8,8,8,28,62,73,28
10200 RESTORE 10210:DIM RI(55,2):FOR A
```

BY PAUL FREEMAN

```
=0 TO 53:READ B,C:RI(A,0)=B:RI(A,1)=C:
NEXT A
10210 DATA 135,186,159,184,175,192,159
,80,127,184,135,192
10220 DATA 79,176,103,160,119,152,135,
88,151,88,127,152
10230 DATA 87,104,119,88,159,120,127,1
84,159,184,103,88
10240 DATA 87,112,103,168,119,128,167,
176,159,96,135,168
10250 DATA 79,120,111,144,95,184,159,1
84,175,128,143,136
10260 DATA 103,120,79,184,111,192,127,
192,135,184,175,136
10270 DATA 79,128,103,184,87,104,127,1
76,159,184,175,128
10275 DATA 87,176,111,192,119,192,159,
160,135,112,87,88
10280 DATA 71,136,95,160,119,192,119,8
8,143,160,167,136
10290 RV=1:P2$(RV)=B2$:RP=INT(6*RND(1)
):RH=RI(RP,0):RV=RI(RP,1):POKE 53250,R
H:RD=INT(3*RND(1))-1
10295 POKE 704,15:SETCOLOR 0,3,6:SETCO
LOR 1,0,10:SETCOLOR 3,8,12:SETCOLOR 2,
8,4
10300 GOSUB 11000:POSITION 0,23:? #6;"
*****":SOUND 0,0,0,0:POKE 53768,4:POKE
 53761,163:POKE 53765,163
10310 POKE 53760,252:POKE 53764,125:GO
TO 23500
10499 REM * SCR 0 *
10500 SCR=1:POP :GOSUB 800:GOTO 100
10999 REM * SCR 1 *
11000 POKE 708,(3*16)+6:POKE 53248,H:P
OSITION 0,0:? #6;"
11005 ? #6;"
11010 ? #6;"
11015 ? #6;"V    XZ  XVVVVZ    "
11020 ? #6;"V        XVZ       "
11025 ? #6;"V                  "
11030 ? #6;"V                  "
11035 ? #6;"V           YjX    "
11040 ? #6;"V           YVuVX  "
11045 ? #6;"V           XVVZ   "
11050 ? #6;"V            VV    "
11055 ? #6;"V            XZ    "
11060 ? #6;"V                  "
11065 ? #6;"V                  "
11080 ? #6;"V                  "
11085 ? #6;"VVVVVVVW           "
11090 ? #6;"WVVVWVVZ           "
11095 ? #6;"V
11100 ? #6;"VWVVVVVW           "
11105 ? #6;"VWuvuuVV           "
11110 ? #6;"VWu1uuvVW j   j    "
11115 ? #6;"VWvuuuuuUQujQSuSjQS"
11120 ? #6;"VWuuuuuvvVUuVVVVuVV"
11200 POSITION 6,23:? #6;"NCOPE ";SC

11300 RETURN
11499 REM * SCR 2 *
11500 POKE 708,(15*16)+8:POKE 53248,H:
POSITION 0,1
11505 ? #6;"
11510 ? #6;"
11515 ? #6;"     XVWWVVZ       "
11520 ? #6;"      XVVVZ        "
11525 ? #6;"       XVZ         "
11530 ? #6;"                   "
11535 ? #6;"                   "
11540 ? #6;"           j j     "
11545 ? #6;"         XVVuWuVVVZ"
11550 ? #6;"          XfftZ    "
11555 ? #6;"                   "
11560 ? #6;"                   "
11565 ? #6;"             Y     "
11570 ? #6;"            YVW    "
11575 ? #6;"           SVVV    "
11580 ? #6;"         jjVVWVV   "
11585 ? #6;"       jSuuVVVWVW  "
11590 ? #6;"      juVVUVVWWVZ  "
11595 ? #6;"    jQuVUUvvvUVV   "
11600 ? #6;"   YuVVUvuv2vvVVW  "
11605 ? #6;"SSQVWVVuvuuvuvvuVVVW"
11610 ? #6;"WWWWWWVUuvvvuvvvuVVV"
11615 POSITION 6,23:? #6;"NCOPE ";SC

11800 RETURN
11999 REM * SCR 3 *
12000 POKE 708,(12*16)+6:POKE 53248,H:
POSITION 0,1
12005 ? #6;"
12010 ? #6;"
12015 ? #6;"          XZ       XVVV"
12020 ? #6;"                   XVW"
12025 ? #6;"                    VW"
12030 ? #6;"                    VW"
12035 ? #6;"                    VW"
12040 ? #6;"        j j         VW"
12045 ? #6;"       XuVuVZ       VW"
12050 ? #6;"     Yj VUUZ       VVW"
12055 ? #6;"     Vu VVZ       VVVW"
12060 ? #6;"     UV VZ       jVVVV"
12065 ? #6;"     VW Z      VuVVVV"
12070 ? #6;"     VW        VUUUUUU"
12075 ? #6;"     VWW             "
12080 ? #6;"     UVZ             "
12085 ? #6;"     VV              "
12090 ? #6;"    YVV              "
12095 ? #6;"   YVUVW     j       "
12100 ? #6;" YVV3VVW   ju  j     "
12105 ? #6;"SVVuuvuVQSuuSQuSQSS"
12110 ? #6;"VVvvuvuvVVUVVUVWWWW"
12115 POSITION 6,23:? #6;"NCOPE ";SC

12300 RETURN
12499 REM * SCR 4 *
12500 POKE 708,(2*16)+6:POKE 53248,H:P
OSITION 0,1
12505 ? #6;"
12510 ? #6;"
12515 ? #6;"VVVZ       XVVUVVZ    "
12520 ? #6;"WVZ         XVVVZ     "
12525 ? #6;"WV           VVZ      "
12530 ? #6;"WV           XZ       "
12535 ? #6;"WV     XVVZ           "
12540 ? #6;"WV      XZ            "
12545 ? #6;"WV              j     "
12550 ? #6;"WV             XVuVZ  "
12555 ? #6;"WV   j          XVZ   "
12560 ? #6;"VV  YuX               "
12565 ? #6;"VZ  XVZ  j            "
12570 ? #6;"Z         u           "
12575 ? #6;"          V         Y"
12580 ? #6;"          V        YV"
12585 ? #6;"         YVW      YVV"
12590 ? #6;"        jVVVj     VVW"
12595 ? #6;"      YQuV4UuQWjjVVW"
12600 ? #6;"  YjSVUvvuvvVVuuVVW"
12605 ? #6;"SQVuVUvuvvvuvvVuVWW"
12610 ? #6;"VVVVVuuvvvvuuvvVWWW"
12615 POSITION 6,23:? #6;"NCOPE ";SC

12700 RETURN
12999 REM * SCR 5 *
13000 POKE 708,(3*16)+10:POKE 53248,H:
POSITION 0,1
13005 ? #6;"
13010 ? #6;"
13015 ? #6;"   XVVUVUVVUUVZ     "
13020 ? #6;"     XVVUVVUVZ      "
13025 ? #6;"      VVVWVVZ       "
13030 ? #6;"      XVVWVZ        "
13035 ? #6;"       VVWV         "
13040 ? #6;"       XVWV         "
13045 ? #6;"        VWV         "
13050 ? #6;"        XVV         "
13055 ? #6;"         XV         "
13060 ? #6;"    j    P     j    "
13065 ? #6;"   juW    3    Yu   "
13070 ? #6;"  YuVZ    V j  XUVV"
13075 ? #6;"SQVVZ    jVUXu   XVV"
13080 ? #6;"VSVZ     uVVVZ    VV"
```

BLACKBIRD

```
13085 ? #6;"VVZ      XZ        VV"
13090 ? #6;"#V                 VV"
13095 ? #6;"#V                 VV"
13100 ? #6;"#VW   j        j  VV"
13105 ? #6;"#VVQQQu5555QjQuQSVV"
13110 ? #6;"#VVVVuuuuvvuuuuuuvVV"
13115 POSITION 6,23:? #6;"NCOPE B ";5C

13300 RETURN
13499 REM * SCR 6 *
13500 POKE 708,6:POKE 53248,H:POSITION
 0,1
13505 ? #6;"UUUUVVVUVUUUUUVVVUUU"
13510 ? #6;"ttrttttrtvvvvvv##vvvr"
13515 ? #6;"         XVVZX##VVZ "
13520 ? #6;"          XVK XVVZ  "
13525 ? #6;"           XVK XZ   "
13530 ? #6;"            XVK     "
13535 ? #6;"    XVUVZ   XVK     "
13540 ? #6;"     XVZ     XZ     "
13545 ? #6;"             VK     "
13550 ? #6;"            VVZ     "
13555 ? #6;"           jVVZ     "
13560 ? #6;"           uVZ      "
13565 ? #6;"  j       VVZ       "
13570 ? #6;"VVuZ    jVVZ      j "
13575 ? #6;"VVZ     uQZ      XuVV"
13580 ? #6;"VV      XZ        XVV"
13585 ? #6;"VV                  "
13590 ? #6;"VV              YK  "
13595 ? #6;"VV              XZ  "
13600 ? #6;"VVW j       j       "
13605 ? #6;"V6VQujSSjSjuSQQQSVV"
13610 ? #6;"VVVVVuUVuVuuvUUVmmmm"
13615 POSITION 6,23:? #6;"NCOPE B ";5C

13700 RETURN
13999 REM * SCR 7 *
14000 POKE 708,(5*16)+10:POKE 53248,H:
POSITION 0,1
14005 ? #6;"UUUVVVVUVUUUUUVVVUUU"
14010 ? #6;"ttrttttrttrrtvvvvrrt"
14015 ? #6;"              XVVZ   "
14020 ? #6;"               XZ    "
14025 ? #6;"                     "
14030 ? #6;"            j        "
14035 ? #6;"          YuKYUVZ    "
14040 ? #6;"          XV##UZ     "
14045 ? #6;"           XVVZ      "
14050 ? #6;"       YKj           "
14055 ? #6;"       XVu           "
14060 ? #6;"        XVK          "
14065 ? #6;"    j   VV       j   "
14070 ? #6;"    uj  VV YVKWKuZ   "
14075 ? #6;"VVVVVuKWVZ XVWXK    "
14080 ? #6;"VVVVVVVVZ           "
14085 ? #6;"     XVZ            "
14090 ? #6;"                    "
14095 ? #6;"           j        "
14100 ? #6;"    j    YuW  j     "
14105 ? #6;"5Q5Qu55jQVvVQ5uQ555"
14110 ? #6;"###vvuuuVu7uUvuvvVV"
14115 POSITION 6,23:? #6;"NCOPE B ";5C

14300 RETURN
14499 REM * SCR 8 *
14500 POKE 708,(12*16)+10:POKE 53248,H
:POSITION 0,1
14505 ? #6;"UUUVVVVUVUUUUUVVVUUU"
14510 ? #6;"ttrttttrttrvvvvvvv##"
14515 ? #6;"            XVWXVVV##"
14520 ? #6;"             XVWXVV##"
14525 ? #6;"              XVWXV##"
14530 ? #6;"               XVWXV#"
14535 ? #6;"          YK    XZYV#"
14540 ? #6;"   YKj  XZ       VV#"
14545 ? #6;"   XVu           XV#"
14550 ? #6;"    XVK           XV"
14555 ? #6;"     XVK   j       "
14560 ? #6;"      XVK  uK      "
14565 ? #6;"       XVHVVZ    XV"
14570 ? #6;" XuZ    XVVZ     YVV"
14575 ? #6;"         XZ     YVVV"
14580 ? #6;"                VVVV"
14585 ? #6;"              jjVVVV"
14590 ? #6;"             jVuuVVVV"
14595 ? #6;"     j       uVVVVVVV"
14600 ? #6;"   jYuW     XZZZZZZZ"
14605 ? #6;"S5QuVuV5jj5WYWYWYWY"
14610 ? #6;"VVVVV8UVuuUVVmmmmmmm"
14615 POSITION 6,23:? #6;"NCOPE B ";5C

14800 RETURN
14999 REM * SCR 9 *
15000 POKE 708,(4*16)+10:POKE 53248,H:
POSITION 0,1
15005 ? #6;"UUUVVVVUVUUUUUVVVUUU"
15010 ? #6;"VVVVVVVVVV9VVVVVVVVV"
15015 ? #6;"VHHHHHHVVVz xVVHHHHHV"
15020 ? #6;"VHHHHVVVz   xVVHHHHV"
15025 ? #6;"VHHHVVz       xVVHHV"
15030 ? #6;"VHHVVz         xVVHV"
15035 ? #6;"VHVVz           xVVV"
15040 ? #6;"VVVz      j      xVV"
15045 ? #6;"VVz       u       xV"
15050 ? #6;"vz        U        v"
15055 ? #6;"          U         "
15060 ? #6;"          V         "
15065 ? #6;"vwj     XVZ      jyv"
15070 ? #6;"VVuj           juVV"
15075 ? #6;"VuQuj          juVuV"
15080 ? #6;"VuuuQuj        juVuuV"
15085 ? #6;"VuuuuQuj       juVuuuV"
15090 ? #6;"VuuuuuQuj      juVuuuuV"
15095 ? #6;"VuuuuuuQu      uVuuuuuuV"
15100 ? #6;"ZZZZZZZZj     ZZZZZZZZZ"
15105 ? #6;"WYWYWYWYu juWYWYWYWY"
15110 ? #6;"VVVVVUVuuuuuuuVUUUUU"
15115 POSITION 6,23:? #6;"NCOPE B ";5C

15300 RETURN
15499 REM * MISSION ACCOMPLISHED *
15500 POKE 53760,110:POKE 53764,55:FOR
 B=0 TO 15:FOR A=0 TO 15:SETCOLOR 4,0,
A:NEXT A:NEXT B
15520 FOR B=4 TO 0 STEP -1:FOR A=15 TO
 0 STEP -1:FOR W=1 TO 5:NEXT W:SETCOLO
R B,0,A:NEXT A:NEXT B
15530 P0$(V)=B2$:P2$(RV)=B2$:POKE 5324
8,0:POKE 53250,0:POKE 53252,0:POKE 756
,224
15540 GRAPHICS 2+16:POSITION 2,2:? #6;
"CONGRADULATIONS!":POSITION 6,5:? #6;"
Mission":POSITION 4,6
15545 ? #6;"accomplished"
15550 POSITION 6,8:? #6;"AND NOW":POS
ITION 5,9:? #6;"ON TO THE":POSITION 4,
10:? #6;"NEXT LEVEL...."
15560 FOR A=0 TO 10:SETCOLOR 0,0,A:SET
COLOR 1,8,A:SETCOLOR 2,3,A:FOR W=1 TO
10:NEXT W:NEXT A
15570 FOR W=1 TO 300:NEXT W:SOUND 0,0,
0,0:SOUND 2,0,0,0:GOTO 10
19999 REM * INTRO *
20000 POSITION 1,5:? #6;"_____
_____":POSITION 1,7:? #6;"_____
_____"
20010 POSITION 5,6:? #6;"blackbird":PO
SITION 3,15:? #6;"COPYRIGHT 1984":POSI
TION 4,16:? #6;"PAUL FREEMAN"
20020 POSITION 0,19:? #6;"PRESS SEL :
LEVEL ";SKL:POSITION 4,22:? #6;"PRESS
 START"
20100 POKE 704,15:POKE 706,60:BH=60:BV
=40:BV2=120:B5=4:CR$=D1$
20110 BH=BH+B5:POKE 53248,BH:P0$(BV)=C
R$:P0$(BV2)=CR$
20115 SOUND 0,BH/7,6,2:SOUND 1,90-BH/3
,10,2:POKE 709,(210-BH)*1.5
20120 IF BH>200 THEN B5=-4:CR$=D2$
20130 IF BH<50 THEN B5=4:CR$=D1$
20140 IF PEEK(53279)=6 OR PEEK(644)=0
THEN GOTO 20500
20150 IF PEEK(53279)<>5 AND PEEK(632)=
15 THEN GOTO 20110
```

BIG /LITTLE BY CARL SCHWARTZ

```
200 R:--------------------------------

201 R: BIG/LITTLE M5PR/DD in  PILOT
202 R:   designed to introduce size
203 R:    and measurement concepts
204 R:            by
205 R:        Carl schwartz
206 R:     in Public Domain  6/84
207 R:--------------------------------

208 J:*HELLO
209 *FILL
210 *MOST
211 *LEAST C:#V=?\2+1
212 C:#A=?\29
213 C:#B=?\29
214 C:#C=?\29
215 C:#D=?\29
216 J(#A=#B)+(#A=#C)+(#A=#D):*MOST
                [+ is a logical or
217 J(#B=#C)+(#B=#D)+(#C=#D):*MOST
218 GR:CLEAR;PEN RED  ;GOTO-75,-20;4(D
RAW30;TURN90);FILL#A
219 GR:GOTO-35,-20;4(DRAW30;TURN90);FI
LL#B
220 GR:GOTO  5,-20;4(DRAW30;TURN90);FI
LL#C
221 GR:GOTO 45,-20;4(DRAW30;TURN90);FI
LL#D
222 T:  1        2        3
4
223 T(#V=1):(least)
224 T(#V=2):(most)
225 A:#F
226 U(#V=2):*BIGGER
227 U(#V=1):*SMALLER
228 J(#T<10)  :*MOST
229 E:
230 *SMALLEST C:#U=1
231 *BIGGEST
232 C:#A=?\7+1
233 C:#B=?\7+1
234 C:#C=?\7+1
235 C:#D=?\7+1
236 J(#A=#B)+(#A=#C)+(#A=#D):*BIGGEST
                [+ is a logical or
237 J(#B=#C)+(#B=#D)+(#C=#D):*BIGGEST
238 U:*DRAW
239 T:  1        2        3
4
240 T(#U=0)*(#V=0)   :(biggest)
241 T(#V=2): most
242 T(#U)0)*(#V=0)  :(smallest)
243 T(#V=1): least
244 *OVER2 T(#E)0): Please try again
245 A:#F
246 *BIGLOGIC
247 J(#U)0)+(#V=1):*SMALLER
248 *BIGGER
249 C(#A)#B)*(#A)#C)*(#A)#D):#R=1
                [logical choice 1
250 C(#B)#A)*(#B)#C)*(#B)#D):#R=2
                [* is a logical "and"
251 C(#C)#A)*(#C)#B)*(#C)#D):#R=3
252 C(#D)#A)*(#D)#B)*(#D)#C):#R=4
253 J(#U=0):*ANS
254 *SMALLER
255 C(#A<#B)*(#A<#C)*(#A<#D):#R=1
                [logical choice 1
256 C(#B<#A)*(#B<#C)*(#B<#D):#R=2
                [* is a logical "and"
257 C(#C<#A)*(#C<#B)*(#C<#D):#R=3
258 C(#D<#A)*(#D<#B)*(#D<#C):#R=4
259 *ANS
260 U(#R=#F):*GOOD
261 C:#E=0
262 C(#R<)#F):#W=#W+1
263 J(#R<)#F):*OVER2
264 J(#T<10)*(#V=0)  :*BIGGEST
265 E(#T<10)*(#V)0)   :
266 J:*DONE
267 *DRAW
268 GR:CLEAR  [drawing module
269 GR:TURNTO 0
270 GR:GOTO-60,-20;DRAW #A*10
271 GR:GOTO-61,-20;DRAW #A*10
272 GR:GOTO-62,-20;DRAW #A*10
273 GR:GOTO-63,-20;DRAW #A*10
274 GR:GOTO-64,-20;DRAW #A*10
275 GR:GOTO-20,-20;DRAW #B*10
276 GR:GOTO-21,-20;DRAW #B*10
277 GR:GOTO-22,-20;DRAW #B*10
278 GR:GOTO-23,-20;DRAW #B*10
279 GR:GOTO-24,-20;DRAW #B*10
280 GR:GOTO 20,-20;DRAW #C*10
281 GR:GOTO 21,-20;DRAW #C*10
282 GR:GOTO 22,-20;DRAW #C*10
283 GR:GOTO 23,-20;DRAW #C*10
284 GR:GOTO 24,-20;DRAW #C*10
285 GR:GOTO 60,-20;DRAW #D*10
286 GR:GOTO 61,-20;DRAW #D*10
287 GR:GOTO 62,-20;DRAW #D*10
288 GR:GOTO 63,-20;DRAW #D*10
289 GR:GOTO 64,-20;DRAW #D*10
290 E:
291 R:--------------------------------

292 *HELLO
293 R: BIG/LITTLE M5PR/DD PILOT PROG
RAM
294 T:HI, I'M A BIG PROGRAM.
295 :
296 :What's your name?
297 A:$NAME
298 GR:QUIT
299 T:
300 T:  LETS GO, $NAME
301 T:PICK A PROGRAM
302 :
303 :1 BIGGEST  Pick the biggest one
 out of 4 choices                (press
 1)
304 :
305 :2 BIG  Which one is big?
                                 (press
 2)
306 :
307 :3 SMALLEST  Pick the smallest o
ne                               (press
 3)
308 :
309 :4 LITTLE  Pick the smaller one
                                 (press
 4)
310 :
311 :5 MEASURE  Use a "ruler" to mea
sure the  length of a line       (press
 5)
312 :
313 :6 MORE/LESS Pick the right one
                                 (press
 6)
314 A:#I
315 M:1,2,3,4,5,6
316 JM:*BIGGEST,*BIG,*SMALLEST,*LITTLE
,*MEASURE,*FILL
317 JM:*HELLO
318 E:
319 R:--------------------------------

320 *LITTLE C:#U=1
321 *BIG          [make up 2 numbers
322 C:#M=?\5+2
323 C:#L=?\5+2
324 J(#L=#M):*BIG [redo if same size
325 C:#N=#L*10
326 C:#O=#M*10
327 GR:CLEAR      [turtle draws..
328 GR:TURNTO 90
329 GR:GOTO-65,-20;DRAW #N
330 GR:GOTO-65,-19;DRAW #N
331 GR:GOTO-65,-18;DRAW #N
332 GR:GOTO-65,-17;DRAW #N
333 GR:GOTO-45+#N,-20;DRAW #O
334 GR:GOTO-45+#N,-19;DRAW #O
335 GR:GOTO-45+#N,-18;DRAW #O
336 GR:GOTO-45+#N,-17;DRAW #O
337 C:#F=1
```

BIG /LITTLE

```
338 *OPS    T(#E)0):try again ,$NAME
339 T:      1                2
340 C(#F=0):##=##+1
341 T(##U=0): BIG (1 or 2) ? \
342 T(##U)0): LITTLE  (1 or 2) ? \
343 A:#F            [accepts answer
344 M(##U=0):1
345 M(##U)0):2
346 CY:##R=#L-##M   [#R if 1 is picked
347 CN:##R=##M-#L   [#R if 2 is picked
348 C:#E=0
349 C(#R(0):#E=#E+1
350 C(#R(0):##=##+1
351 J(#R(0):*OPS   [jump if wrong
352 J(#F=0):*OPS   [or return only
353 U:*GOOD
354 J(#T(10):*BIG
355 R:-------------------------------
356 *DONE SO:#Z
357 C:#Z=(10-##D*10
358 T:$NAME, MISSED ##W and got #Z %RI
GHT \
359 U(##W()0):*LESSON
360 T(##W=0): YOU REALLY MEASURE UP $N
AME
361 PA:150
362 J(##W=0):*REWARD
363 C:#Q=#Q+1
364 E:
365 R:-------------------------------

366 *GOOD J(#T)9 ):*DONE
367 T:GOOD ANSWER
368 C:#T=#T+1
369 C:#S=0
370 *LOOP SO:#S
371 ,    C:#S=#S+1
372 ,    J(#S(31):*LOOP
373 SO:
374 E:
375 R:-------------------------------

376 *MEASURE E(#Q)0):
377 C:##=?\6+1
378 C:#O=##*15
379 GR:CLEAR
380 GR:TURNTO 90              [horiz
381 C:#V=?\7*10-15
382 GR:GOTO-65,#V ;DRAW #O
383 GR:GOTO-65,#V+1;DRAW #O
384 GR:GOTO-65,#V+2;DRAW #O
385 *REDO E(#Q)1):
386 T: HOW BIG (1 2 3 4 5 or 6)
387 A:#A
388 M:##
389 C(#A()#M):##=##+1
390 UY:*GOOD
391 JN:*TRYMORE
392 J(#T(11):*MEASURE
393 E:
394 *TRYMORE  T:TRY AGAIN $NAME
395 PA:30
396 J:*REDO
397 R:-------------------------------

398 *REWARD  GR:QUIT
399 T:OK, $NAME, LETS PLAY A DRAWING G
AME
400 :
401 :
402 : PICK A NUMBER  1, 2, OR 3,$NAME
403 T:          MENU
404 :
405 :1  *TREE
406 :
407 :2 *SPIRAL
408 :
409 :
410 :3 *TURTLEDRAW
411 :
412 :4 *SURPRISE
413 :
414 T:
415 :
416 : 1TREE  2SPIRAL  3DRAW TURTLE
417 C:#Q=#Q+1
418 T:PICK THE NUMBER OF YOUR CHOICE
419 A:#C
420 J(#C=2):*SPIRAL
421 J(#C=3):*TURTLEDRAW
422 J(#C)3):*SURPRISE
423 R:-------------------------------
424 *TREE C:#P=?\12
425 GR:GOTO -1,-31;TURNTO 0;PEN RED
426 GR:DRAW 32;GOTO1,-31;DRAW32
427 GR:GOTO  0,-31;TURNTO 0
428 T:          $NAME'S  tree     \
429 U:*LESSON
430 C:#L=2*2*2*2*2*2
431 *BRANCH SO:#L
432 C:#B712=#P*16+1  [Poke to change c
olor
433 C:#L=#L/2
434 GR:DRAW #L
435 R:GR(#L(3):PEN BLUE
436 GR:TURN -45
437 PA:5
438 U(#L()1):*BRANCH
439 GR:TURN 90
440 U(#L()1):*BRANCH
441 GR:TURN-45;DRAW-#L
442 C:#L=#L*2
443 E:
444 R:-------------------------------

445 *SPIRAL GR:CLEAR;PEN BLUE
446 T:TURN (0-22)?\
447 A:#A
448 T:      $NAME'S SPIRAL  TURN #A
\
449 U:*LESSON
450 C:#D=0
451 GR:GOTO 0,10
452 *DRAWLOOP
453 SO:#D
454 C:#D=#D+1
455 GR:DRAW #D;TURN90-#A
456 J(#D(70):*DRAWLOOP
457 E:
458 U:*LESSON
459 R:-------------------------------

460 *TURTLEDRAW  GR:CLEAR
461 C:#P=0
462 T:INPUT TURN ANGLE (0-89)\
463 A:#T
464 T:INPUT TURTLE LENGTH (9-99)\
465 A:#D
466 T:INPUT TURTLE TRIPS (22-99)\
467 A:#R
468 *REPETITION   SO:#C,#P
469 C:#C=?\3     [random colors
470 GR:PENYELLOW
471 GR(#C=1):PEN BLUE
472 GR(#C=2):PEN RED
473 GR:DRAW#D;TURN180;PEN UP;  DRAW#T;
TURN180
474 GR:TURN 90;TURN#T;DRAW 1
475 C:#P=#P+1
476 J(#R)#P):*REPETITION
477 T:$NAME'STURTLE #TTURN ##DLONG #RTR
IP \
478 U:*LESSON
479 E:
480 *SURPRISE [Insert you'r own reward

481 T:%
482 POS:5,5
483 T:PICK ANOTHER NUMBER, $NAME
484 PA:100
485 J:*REWARD
486 *LESSON SO:
487 T(#I=6):Most/Least\
488 T(#I=1):Big\
489 T(#I=2):Biggest\
490 T(#I=3):Small\
491 T(#I=4):Smallest\
492 T(#I=5):Measure\
493 E:
```

The 'C' Duffer

"You should learn C" my Atari Guru said; "It's almost as fast, flexible and powerful as machine language, and nearly as easy to write as BASIC." So I took Larry Gold's suggestion and got ACEC, by ace programmer Ralph E. Walden, from Librarian Ron Ness (374 Blackfoot, Eugene, OR 97404).

And it was true: C is fast (12x faster than Atari BASIC and 7x faster than IBM PC BASIC, running BYTE mag's 'Sieve' benchmark), flexible, powerful, and easy to write — ONCE YOU KNOW HOW. It's also frustrating to learn; none of the books on C really apply to Atari, with its special I/O.

'C' is like a race car engine which you install in the family car. It's a C-engine, more or less, but the installation is different in different cars. And there is no manual for sports cars — like Atari. You gotta learn from your mistakes. Or better, from my mistakes, which is why this is being written.

It took most of my computer time for a week or more before I could even see any output from a C program. Not just those I wrote, but even the demo on the ACEC disk produced no output on my screen.

Lesson #1: if you get DOS menu's instead of answers try getkey() as a final command to hold the data on your screen. Otherwise it can be on, and off, in one scan cycle of 1/60 sec!

Most of the mystery for C duffers is in I/O, like how to get a simple number into your Atari, and then get something recognisable back out. The full-bore C's have scanf(), a command or function which accepts formatted input. We don't have this, tho you could write the code for this and add it to your own C — it can be customized, like FORTH. We have to work with simpler tools, at least initially. Functions like getkey() getchar() or gets() which read in a keystroke, character code or string respectively are rather like BASIC's GET or INPUT commands.

Here is one of the first C routines I wrote which actually let me enter numbers, do something with them, and then spit out a result:

```
10 /* READ n NUMBERS, SUM THEM */
20 /* AND TAKE THEIR AVERAGES  */
30 /*    WITH INTEGER MATH     */
40 /*                          */
50 main()  $(
60 char *str;  /*A char.string*/
70 int i,a,b,c;/*and 4 integers*/
80   a=b=c=i=0;/*are declared.*/
90   for(;;) $(  /*Endless loop*/
0100   printf("Enter #: ");
0110   gets(str); /*Enter a no.!*/
0120   a=a + val(str),++i; /*sum*/
0130   b=a/i; /*integer division*/
0140   c=a%i; /*get remainder*/
0150   printf("sum= %d",a);
0160   printf("  n= %d",i);
0170   printf(" ave= %d",b);
0180   printf(" mod= %d\n",c);
0190  $)
0200 $)
```

The remarks between the /* */ pairs are not part of the code, and if you copy and compile this mini-program they can be left out. But be careful not to leave out anything else, like a ';' or you'll get an error message (not just a number as in BASIC) when you compile.

This source code was written with Atari's Assembler-Editor. One ACEC's several good points is it can accept line-numbered code. You c also use Atariwriter (no line numbers) or other text editors, but you must avoid header info which will confuse the C compiler. To use Atariwriter I start with a dummy line loaded from disk, which avoids the header problem:

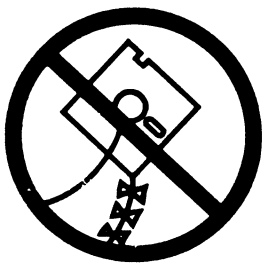
My ugly duckling program let me experiment with the keyboard input process; to my surprise ACEC will let you enter a number like 1.23456E3. It's a surprise because on many big C machines it takes a special bit of programming to make this notation work. Using integer math, fractions are dropped so the above number enters as 1234. Next time we will use another of ACEC's goodies, floating point math, which many bigger C's don't provide. See you next month!

— Dick Barkley, Acting President

(He was nominated at our last meeting. As our former president had to resign for personal reasons, and with the concurrence of our vice president, Dick Barkley is "acting president" until the election at the May meeting. — Ed.)

BLACKBIRD

```
20152 SKL=SKL+1:IF SKL)3 THEN SKL=1
20155 POSITION 18,19:? #6;SKL:GOTO 201
10
20500 P0$(BV)=B$:SETCOLOR 1,0,0:SETCOL
OR 0,0,0:SETCOLOR 3,0,0:SETCOLOR 2,0,0
:CR$=D1$:SOUND 0,0,0,0:SOUND 1,0,0,0
20510 RS=SKL+3:RETURN
21999 REM * END *
22000 FOR A=0 TO 8:POKE 704,(2*16)+10-
A:POKE 712,14:V=V+3:POKE 53256,1:P0$(V
)=D3$:FOR W=1 TO 5:NEXT W:P0$(V)=D1$
22010 POKE 712,0:POKE 53760,50+A*15:PO
KE 53761,14-A:FOR W=1 TO 3:NEXT W:POKE
 53256,0:P0$(V)=D2$:FOR W=1 TO 3
22015 NEXT W:NEXT A
22020 POKE 53761,4:FOR A=V TO 210 STEP
 3:POKE 712,14:POKE 53256,1:P0$(A)=D3$
:FOR W=1 TO 5:NEXT W:POKE 712,0
22030 P0$(A)=D1$:FOR W=1 TO 3:NEXT W:P
OKE 53256,0:P0$(A)=D2$:FOR W=1 TO 3:NE
XT W:NEXT A
22040 SOUND 0,0,0,0:P0$(A)=B2$:P2$(RV)
=B2$:POKE 53248,0:POKE 53250,0:POKE 53
252,0:FOR W=1 TO 300:NEXT W
22050 POKE 756,224:GOTO 10
22999 REM * PAUSE *
23000 SETCOLOR 4,10,0:POKE 764,255:FOR
 W=1 TO 100:NEXT W
23010 IF PEEK(644)=0 OR STICK(0)<>15 O
R PEEK(764)<>255 THEN SETCOLOR 4,0,0:P
OKE 764,255:GOTO 105
23020 GOTO 23010
23499 REM * INT FREEZE *
23500 P0$(V)=CR$:POKE 53248,H:POKE 532
78,1:GOTO 23010
```



FID CON'T

```
    addr(a)=0
    OD
  Graphics(0)
  FOR a=0 TO 65
    DO
    IF mark(a)=1 THEN
      Print("Reading File:  ")
      c=directory+length*a
      Get_filename(c)
      Readfile(a)
      IF flag=1 THEN a==-1 flag=0
    FI
    OD
  Dump(a)
RETURN

PROC FID()
  BYTE key=764
  DO
  key=255
  Menu()
  WHILE key=255
    DO
    Blip()
    FOR c=1 TO 8000
      DO
      OD
    OD
    choice=GetKey()
    choice==-48
    IF choice=5 THEN Checkformat()
    ELSEIF choice=1 THEN Copy()
    ELSEIF choice>1 AND choice<5 THEN
      Execute()
    FI
  OD
RETURN
```

ACTION! ERRORS?

(reprint: SBACE, April 1985)

ACTION! has some very powerful error trapping capabilities of which I am not sure everyone is aware. This information is in the Action! user manual, if you recognize it when you see it.

In BASIC you have the TRAP statement so when an I/O error occurs, you can divert the program in an attempt to recover from the error.

In Action! there is no TRAP statement or KEY WORD. Action! takes a more direct and flexible route to accomplish the same thing (see section VI.7.3 on PROC Error).

There is a vector through which the Action! program jumps whenever there is any kind of I/O error. The result is to return you back to the Action! monitor. All you need to do is change the vector to point to your own error procedure.

Example:

```
PROC indaterr(BYTE errcode)
;
;this will declare your
;error routine. errcode
;is the variable which is
;passed to your routine
;to tell you what error
;has occured
;
;your code goes here
;
RETURN ;gets you back to
;where the error
;occured
PROC indat()
;
;this is your routine for
;getting data. indaterr
;will handle any errors which
;occur here

CARD temperr ;hold the old
;error address so it can
;be restored when you no
;longer need to trap in
;indat
temperr=Error ;this saves the
;old error handling addr
Error=indaterr
;this will set the error
;vector to point to your
;error routine

;the rest of your code
;goes here
;
Error=temperr
;
;this restores the old
;error vector
;this way you can have
;many error trapping
;routines to handle
;different PROC and also
;you must restore this
;to its original value
;before returning to
;the ACTION! monitor
;
```

— J. "Polymorphic" Patchell†

BUMPAS REVIEWS

Super-Text Professional Word Processor (MUSE Software, 347 N. Charles St., Baltimore, MD 21201) is a full-function word processor which has recently been gaining more attention. We're familiar with MUSE Software because of their Wolfenstein series of action adventure games. With Super-Text, they have given us a good application.

The title menu when booted up gives the user the option of using Atari DOS to copy a disk or perform formats, etc. From this menu one may also run a "System Options" program in order to modify printer parameters for the printer you use, and to change editing defaults. The first time I tried to set the parameters to match my IDS, I failed. But I find the default Epson parameters print the text just fine. I just cannot underline or use boldface.

There are some interesting features here which are not found in other word processors I have seen. A string of up to 30-characters can be defined and stored in a buffer which is recalled by "Control-'". This can save a lot of typing if you have a phrase which occurs often in a text file you're preparing. The default string in this buffer is "the".

A HELP module can be loaded into memory for on-screen help information on any one of ten topics. The 14k text buffer is shortened when this option is used, however. Search and replace functions are very much more powerful than in other word processors for the Atari. These searches may be performed either forward or backward through the text. The characters "!" and "&" may be used as wildcards (for characters and spaces) in a search.

In addition to the common block operations, the addition of the ability to write a defined block to a disk file, or append it to an existing disk file is a very powerful feature. Text deletions are also expanded in this powerful program. A line delete occurs from the cursor to the end of the line. The user may delete individual characters and words as well as the whole screen from the cursor to the end. A 3-key combination will delete the entire file from memory.

The cursor can be moved to any of the 4 screen edges as well as the normal Atari 4-direction cursor movement. Two key-strokes will take you to the beginning or end of the file. One key-stroke will show information on the current status of Super-Text. Here you may check to see if the Help and Print modules are in memory. You can also toggle and define the character string to be used in "The" Key, as well as toggle "Autolink" on and off. The screen will also show if you've made any changes to the current file and how many strings were replaced in your last Replace operation. The number of bytes used in the file, and the number of free bytes remaining will be shown, along with the filename. Form and sheet feed may be toggled from this menu without having to re-enter the System Options program. The current file may be previewed here to see how it will appear on a printed page.

The Autolink function is a powerful one and helps the user overcome some of the limitations of the rather small 14k text buffer. If this function is toggled on, the following functions will be performed across multiple files automatically, with no manual intervention: Find, Replace, Print, and loading the next linked file. A "link" is a filename enclosed by "CTRL-Q"s and bars (SHIFT-=). Links must occur at the top or bottom of a file. This position must correspond with the direction indicator (+ or -); i.e., if at the top, the direction must be "-". There may not be even a carriage return between the link and the next file. Autolink will function across multiple disk drives, too.

Everytime you want to load a file into memory, the directory of the selected drive is shown on the screen. D1: is the default, but this may be changed. To load a file from the directory, you need only select a number displayed next to the filename. Files may also be saved or deleted with one key stroke in the same way.

Multiple line headers and footers may be printed on one page, all pages, or on alternating pages. Page numbers may be in the format of "CTRL-Nc-p", where "c" is the chapter number, and "p" is the page number. A wire-bound manual with more than 150 pages and a Quick Reference card are provided. The program comes with two system disks, in case you wear one out. If both program disks become damaged, replacements may be ordered for $10. If you don't already own a full-featured word processor, ask to see this one demonstrated at your local store.

MAY MEETING
WED. MAY 8TH

7:30PM
SOUTH EUGENE HIGH

BREAKTHROUGH IN THE ARDENNES ($40, SSI) is a smootly executed simulation of the last major battle on the Western Front during WW2. This is a "classic" simulation, war-gamed by dozens of paper board games over the years. The computer version permits both solitaire and two-player enjoyment. Each turn is a day of game time. Three scenarios include the first and last 7 days, as well as the whole 12 days.

The map scrolls over several screens containing a 32 by 31 hex grid, each of which "represents an area of about two miles". Parts of Germany, France, Belgium and Luxembourg are included. Terrain includes clear, rough, forest, town, river and road. The depiction of roads is unusual — they are indicated by white hexsides. Bridges are a very important terrain feature, but they are indicated on the screen only by the white hexside of a road. When the bridge is destroyed, it must be rebuilt by the player desiring to use it. Players must refer to a plastic-coated play-aid map to see bridge sites which may be rebuilt. Only Engineer units may rebuild bridges for the Germans.

Maneuver elements include battalions, regiments and divisions. Units belonging to the same formations may be combined or broken down into separate units. While there are only 12 turns, each turn has 12 phases. The program performs very quickly, and players are offered the option to save a game in progress at several points during a turn.

Tracing supply lines to a friendly edge of the map is crucial to success in the game. Air points are available for ground attack and interdiction roles. Artillery and replacement points may be added to ground units. Units may enter a "Travel" mode for more rapid movement along roads. Units may also fortify positions. Combat is plotted with the option to advance into a vacated enemy hex. Weather may affect the movement of units.

The 5 levels of victory conditions are based upon the possession of towns on the map. The 7 pages of documentation include a historical commentary, notes on strategy and tactics, an Order of Battle Chart, and some examples of play. A novice to the hobby of playing historical simulations might enjoy this one. For those of intermediate or expert skill, this game provides subtle challenges.

VISUALIZER the graphics management system and electronic slide creator/projector ($50, Maximus, Inc.) doesn't sound like another drawing program — and it isn't. The name indicates what it has which is lacking in other drawing programs. The user can create and save graphics screens of one's own design.

In addition, text may be put right into the design of the screen, or in a caption. In addition to normal Atari text, one may select from among Tall, Wide, Italic, Shadow and Striped text. Upper and lower case and Inverse text is avaiable in all modes. You can also use a character editor (not provided) to create a custom text font for your use.

Drawing is done by joystick, and the text window shows a menu of functions: Animation, Border, Circle, Diagonal Drawing, Erase, Fill, New Colors, Load Slide (screen), Merge Slide, New Font, Oval, Paint, Rectangle, Save Slide, Text (in graphics window), Utilities (including a disk directory of slides and character sets, rename file, delete file, format disk), Caption (add text window to screen), run Slide Show program.

The spacebar toggles one of two menus on the screen at any time while drawing a screen. A Help function will give you abbreviated instructions at any time.

The Show Slides program is where this package really shines. The menu on this program provides an exit back to the drawing program as well as the ability to sort a directory list of screens into the desired order and show them. This presort file may be saved. The pace of the slide show can be set with a timer function for automatic running. The user may manually show the slides if desired. When the automatic timer is running, any slide can be held on the screen for extended viewing. An arrow pointer can be toggled on and off to accent a slide presentation.

The animation function of the drawing program offers 6 options: switching 1, 2, or 3 colors; Marqee; Sparkle; and Rainbow. These last three options provide different ways to rotate colors on a screen. There is a jigsaw game which scrambles a slide into 25 pieces which you reassemble with the joystick. The score depends upon elapsed time and the number of pieces correctly placed. SELECT shows the solution, but speeds up the clock; OPTION unscrambles the picture automatically. START restarts the game.

The program provides the ability to print screens to Epson or C.Itoh type printers. The Visualizer package comes with a cassette tape and instructions for synchronizing audio effects or narration to the slide show. This requires an audio cassette recorder and a specially designed cable, as well as an Atari 810 or 1010 program recorder. Instructions for making the cable are included.

The 40-page manual provides a BASIC subroutine for using a screen in a program of your own design. There is an 800 number to call, a Consumer Hotline. The user may also obtain a data cable for synchronizing narration, instead of making it oneself, by calling this number.

Most drawing programs are nice, but you are left with your drawings and nowhere to go with them. This program is as powerful as any drawing program, but gives you several practical things you can do with the screens once drawn.

HOMEPAK ($50, Batteries Included) gives the user the most value per dollar for any home applications package I've seen. It's three programs integrated into one menu system. HomeText, HomeFind and HomeTerm may all share the same data files. All three programs use a custom character font for displaying text which is pleasingly different from the usual Atari set, and yet easy to read. Multiple drives are supported, and files can be read and written on double density.

HomeText is an amazingly powerful word processor offering many features found on more expensive packages. The screen in the editing mode defaults to a pleasant green, but the screen color can be changed. In addition to the Atari cursor arrow controls, you can quickly move to top or bottom of the text file. Two key-strokes can move the cursor to the top, bottom and middle left of a screen, as well as to the beginning or end of a line.

The Atari key toggles the insert mode, so inverse characters don't seem to be available. Blocks can be moved, copied and deleted. Search and replace functions are implemented. A print preview option enables the user to see how the text will appear on a printed page — sort of. The screen shows dots and colors to show the location of lines of text and whether they are bold, extended or underlined.

The text buffer is less than 7k, so there's not a lot of space in which to work. But an "include" function can link together more than one file for printing. There's an elegant way to use unsupported printers with this program. A menu window pops on the screen into which you type the decimal ASCII code for feature desired. The program will print multiple line headers and footers, with page numbers.

This program is excellent for the occasional user of a word processor who might forget, from time to time, the commands to use. Well, you don't have to go back to the documentation for another course in word processing. The program is menu driven at every step of the way. For someone using the program every day, these menus might be bothersome. The documentation explains how to short-circuit some of the menus, so the authors recognize the problem.

HomeFind is a filing program which doesn't require the user to study a manual to learn how to set up database fields. Data is entered in a common-sense manner upon what looks like an editing screen in a word processor. Each entry can contain up to four elements which total less than 144 characters. No element may contain more than 80. This is generous enough for most small filing jobs such as mailing lists. The only concession to "fields" one must make is the use of 's after two of the elements, and a "bar" to denote a comment field. An apostrophe may be used to replace an entire string ending in 's to repeat a previously used string.

Data may be recalled merely by typing a string (less the 's) equal to any of the first three elements in an entry. Multiple criteria may also be used. Data files to and from HomeText may be merged. HomeFind data disks may not contain any other files.

HomeTerm is the star of this package. This communications program is as good as any on the market for the Atari. It supporst 1200 baud. In addition to the usual communications modes (ASCII, ATASCII and XModem), it also supports the Videx mode used on CompuServe. The default green screen color may be changed. Text and screen brightness may also be controlled.

When text is displayed on the screen, a word-wrap function may be toggled on and off. Word-wrap makes the text easier to read since lines won't break words up in the middle. An editing window can be displayed during communication. This window will hold up to 120 characters which will not be sent to the other computer until you press RETURN. This is handy when in a chat mode, or when using the CB simulator or conference functions on CompuServe. Without it, sometimes you'll miss what others are saying while you're typing a response.

You can open a "capture" buffer with the OPTION key while on the screen displaying the data you want to capture. You don't need to escape to some menu to open the buffer. Handy DOS functions are accessible from within HomeTerm to read directories, delete, rename, lock & unlock files, copy, and format disks. A 24 hour real-time clock can be set to keep track of on-line time. "Macros" can be written and even linked together to automate dialing and logging sequences. Configuration files can be saved to disk which contain the current list of macros, as well as choices of baud rate, screen color and brightness, key click, duplex mode and translation type.

The documentation describes how to use the Atari 835/1030 and MPP modems. There is also an 8-page article by Ron Luks of CompuServe describing how to use CompuServe's SIG*Atari. HomeTerm alone makes HomePak worth the purchase price, in my opinion.

— Jim Bumpas, Co-Editor

Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to Atari Corp. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are $14 a year for U.S., and $24 a year Overseas Airmail and include about 10 issues a year of the ACE Newsletter.
Subscription Dep't: 3662 Vine Maple Dr., Eugene, OR 97405.

**Acting President-Dick Barkley, 2907 Wingate, Eugene, OR 97405 503-344-5843

Vice Pres-Larry Gold, 1927 McLean Blvd., Eugene, Or 97405 503-686-1490

Librarians-Ron and Aaron Ness, 374 Blackfoot, Eugene, Or 97404 (503)689-7106.

Editors— Mike Dunn, 3662 Vine Maple Dr., Eugene, Or 97405 503-344-6193

Jim Bumpas, 4405 Dillard Rd., Eugene, Or 97405 503-484-9925

E.R.A.C.E. (Education SIG Editor)-Nora Young, 105 Hansen Lane, Eugene, Or 97404 /503-688-1458

Send 50c stamps or coin ($1 overseas) to the Ness' for the new, updated ACE Library List-new in May 84 !

Bulletin Board
(503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400 with a TARA keyboard, 2 double-density double sided disk drives, an Epson MX80 printer, a MicroBits 1100E Modem; running the FoReM Bulletin Board software distributed by MicroBits.

Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct 81 to June of 1982

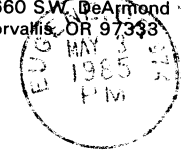
Volume 2 covers July 1982 to June 1983

Only $12 each ($2 extra for Airmail). Available only from:

George Suetsugu
45-602 Apuapu St
Kanoehe, HI 96744

SortFinder 1.2
A composite index of Atari related articles from 5 popular computer periodicals from Apr '81 to June '83, including ACE. Only $6 for ACE member from:

Jim Carr, Valley Soft
2660 S.W. DeArmond
Corvallis, OR 97333



ATARI COMPUTER ENTHUSIASTS
3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL